應用動態緩衝區提升同儕式網路隨選視訊系統之擴充性

Applying Dynamic Buffering to Enhance the Scalability of VoD System on P2P Network

林朝興
南台科技大學 資訊管理所
mikelin@mail.mis.stut.edu.tw

王亮淳
南台科技大學 資訊管理所
m9590101@webmail.stut.edu.tw

摘要

以同儕式網路(Peer-to-Peer network)為基礎的隨選視訊服務，透過資源共享的模式，由系統內的節點提供新進節點影片片頭資訊，因此，節點如何有效的保留片頭資訊是一個關鍵的議題，其影響節點服務能力的堅韌度(robustness)及整體系統的擴充性(scalability)。本篇論文使用多重描述編碼(Multiple description coding)技術對影片串流進行編碼，讓使用者可依自身需求或頻寬限制調整影片觀看品質，並同時考慮系統中資源的平衡。動態緩衝區是利用多重描述編碼的特性，讓節點動態調整緩衝區內所暫存的影片內容，將影片片頭延伸至節點緩衝區所能提供的最大長度，增加等待服務新進節點的時間。在實驗結果中顯示動態調整緩衝區適用於隨選視訊服務中不同熱門程度的影片，有效利用節點儲存空間的資源且降低伺服器端的頻寬耗用，使系統有能力服務更多的節點需求。

關鍵詞：*同儕式網路，隨選視訊，多重描述編碼，動態緩衝區，擴充性*

Abstract

The purpose of this study is to investigate dynamic buffering mechanism for Video-on-Demand (VoD) streaming on Peer-to-Peer (P2P) network. In general, peers have sufficient storage and bandwidth to cache and deliver the initial part of the video to other peers. The key challenge is keeping the initial part of the video in a peer's buffer as long as possible, which affects the robustness of service capacity of a peer and the scalability of the VoD system. Thus, we apply multiple description coding (MDC) technique which provides various quality of the video according to the limitation of peer bandwidth. Moreover, in order to more effectively use resource, we also address the issue of resource balance and propose *Dynamic Buffering* mechanism. It relies on the feature of MDC to dynamically adjust the content of the video cached in a peer's buffer. It is crucial to extend the available time of the initial part of the video and thus to increase the servicing time for a new peer. Our simulation experiments show that the Dynamic Buffering is suitable for various popular degrees of the videos. It efficiently utilizes storage resource of peers, reduce server bandwidth consumption, and maximize the number of served peers.

Keywords：*peer-to-peer network, video-on-demand, multiple description coding, dynamic buffering, scalability.*

一、緒論

隨著網際網路技術的蓬勃發展及使用者對多媒體資訊的需求增加，使得網際網路的應用從Web瀏覽轉向以多媒體內容爲中心的綜合應用。近年來，同儕式網路(Peer-to-Peer Network，P2P Network) [2][5][8][15]快速發展，不同於主從式架構(Client-Server model) [6][10]，其作法是透過各個節點彼此合作以存取影片內容。在同儕式網路的使用者稱之為節點(peer)，每個節點可同時扮演「伺服器端」(server)和「客戶端」(client)的角色，節點的資源共享有助於節省伺服器端頻寬及儲存空間的耗用，改善系統整體的處理能力及負載能力。因此，隨著網路上節點數量的增加，相較於主從式架構，同儕式架構能有更佳的系統擴充性(scalability)。

同儕式網路及多媒體串流傳輸技術的結合成為熱門的網路多媒體串流傳輸架構[3][5][11][12]。目前以同儕式網路為基礎的多媒體串流服務可區分為下列兩種模式：

- **即時串流(Live Streaming)**。即時串流以傳送即時性的多媒體內容為主[9] [13]，例如：線上新聞、演唱會現場直播。使用者可以直接上線觀看即時且同步的影音內容，完全沒有空間上的限制。由於即時串流的服務內容具有即時性，使用者關心的是目前正在播放的影片內容，不會請求在加入時間點之前所播放過的影片內容，因此即時串流的模式中，新進節點不會向系統提出服務影片片頭的請求。
- **隨選視訊(Video-On-Demand, VoD)**。隨選視訊技術[1][3][6][10][14][16]是以預錄且不具即時性的多媒體內容為主，將其放置在網路上進行播放。不同於即時串流，隨選視訊的服務重點為提供客戶端「完整」的影片內容，客戶端對於影片的觀看需求是從影片開始至影片結束，故影片片頭的取得成為其服務的重點。一般在同儕式網路上提供隨選視訊服務的作法是透過節點將部分影片內容暫存(cache)於緩衝區(buffer)，當有其它節點提出請求時再傳送，此種方法稱之為cache-and-relay，其可節省伺服器端頻寬及儲存空間的耗用。

本篇論文將探討建置於同儕式網路上的隨選視訊服務，透過節點影片資源共享，可有效降低伺服器負載，但仍存在一些待改善的議題：

- **節點的服務能力受到緩衝區空間的限制。**節點會在任一時間點向系統提出服務請求，稱之為非同步(asynchrony)請求。由於節點受限於緩衝區所能儲存的空間，當節點等待新進節點的請求時間過於冗長，緩衝區空間已滿載，節點將不再持有影片片頭，也就失去服務新進節點的能力。此時，新進節點將會轉向伺服器端請求服務，增加伺服器端的負載，系統為了服務新進節點的非同步請求將導致伺服器端負載過重。因此，隨選視訊的服務必須透過節點利用有限緩衝區空間，並有效的暫存影片片頭資訊，以服務新進節點[3][5][11]，提高系統擴展性。
- **隨選視訊的服務須仰賴龐大且長時間的頻寬支援。**節點對於影片串流的存取時間具有連續性，若不能有效的運用節點資源，大量的客戶端非同步需求將導致伺服器端負載過重，降低系統服務的擴展性。

有鑒於目前在同儕式網路上提供隨選視訊的服務仍存在上述的缺失，因此本論文提出動態緩衝區(Dynamic Buffering)－以CoopNet[13]系統架構與多重描述編碼(Multiple description coding, MDC)[4][7][12][13]串流編碼技術為基礎。MDC可將影片串流編碼成多條子串流，節點只要任意接收其中一條子串流即可觀看影片，隨著接收的子串流數量增加即可改善影片品質。同儕式網路上節點的連線品質不一，例如：ADSL, Cable Modem...等等，由於節點的頻寬受限，將影響節點對於觀看品質的需求，此特性稱之為「連線異質性」(heterogeneity)。在本篇論文中，使用MDC改善此議題，提供多種不同位元率的影片版本，以適應網路上節點的連線異質性及滿足不同觀看品質的需求。

動態緩衝區的作法是透過MDC將影片串流進行編碼，當節點緩衝區中即將失去影片片頭的資訊時，漸進式的調整節點緩衝區內所暫存的子串流數量，使更有效率的運用，動態調整節點緩衝區之子串流，主要目的為**提高節點可用率(availability)**，**降低節點服務的拒絕率，減少節點向伺服器端請求的次數及所付出的頻寬資源**。

本論文的架構如下：第二章節為相關文獻探討，第三章節介紹動態緩衝區機制在隨選視訊服務上的應用與所需假設及方法，第四章節為動態緩衝區的效能評估，最後，第五章節為本論文的結論。

二、文獻探討

2.1 多重描述編碼

MDC[4][7][13]編碼技術的作法是將原始的影片串流資料編碼成數條影片品質相同的串流，每一條稱之為description， description之間不具相依性且可獨立解碼，節點只要接收到任意數量的description即可進行解碼。MDC編碼技術同時考慮節點對於影片觀看品質的需求及頻寬的限制，低頻寬使用者受限於傳輸能力可選擇接收部份的description，取得具有一定失真率的影片觀看品質。反之，高頻寬使用者可透過增加接收的description數量以取得較佳的影片觀看品質，其能滿足同儕式網路上節點的特質，提供數種不同位元率的多媒體串流。

2.2 CoopNet

CoopNet[13]結合主從及同儕式兩種網路架構。CoopNet 採用中央集權的管理方式，由伺服器端管理節點，使節點能快速的加入系統以取得影片服務。串流的傳送是以應用層群播樹(application level multicast tree)[5][7][9]為系統主體架構，其作法是由節點形成一個樹狀結構的網路，將原始影片編碼過後所產生的 description，透過伺服器端分配到多棵樹中進行群播，並透過節點執行封包的複製及繞送程序。CoopNet 為延伸系統擴展性，允許節點向數個父節點請求服務，透過節點之間的互助使其達到觀看品質的需求，彼此共享 description 資源，使 description 資源的運用與取得更有效率。

然而，以 CoopNet 為基礎的隨選視訊服務衍伸出以下相關議題：

- **節點的緩衝區空間受限**。隨選視訊服務必須透過節點暫存部份的影片區塊(segment)，以降低伺服器端的負載。當節點的緩衝區空間受限時，其服務新進節點的能力相對受到限制。
- **資源不平衡**。伺服器端將影片串流使用 MDC 進行編碼，將 description 分配給不同的節點，但當網路上的 description 數量不平衡時，可能導致一個新進節點無法得到滿足的觀看品質，此時將由伺服器端進行服務，增加伺服器端的負載。

本論文為改善上述的議題，提出動態緩衝區及資源平衡的概念。動態緩衝區是以動態的方式調整節點緩衝區內 description 的數量，同時考慮系統中的 description 資源分布情形，平衡每條 description 在網路上的數量，使滿足不同新進節點對於影片觀看品質的需求，減少向伺服器端請求服務的次數。

三、動態緩衝區

動態緩衝區是應用於同儕式網路架構的隨選視訊服務。動態緩衝區的提出是為了解決節點受限於緩衝區空間大小，而無法提供新進節點服務的問題。藉由動態緩衝區可使系統中仍然持有影片片頭的節點，動態的調整緩衝區內暫存的影片片頭長度，延長等待新進節點加入的時間爾後進行服務。

此章節將詳細探討動態緩衝區在隨選視訊的應用，以及如何輔助伺服器端及客戶端的多媒體影片串流資源配置。隨選視訊的服務重點為提供節點「完整的影片內容」，因此影片片頭的取得即成為系統的服務關鍵。一個完善的隨選視訊系統，必須

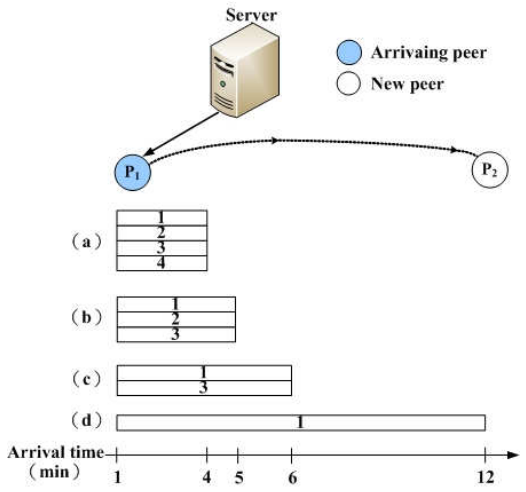

圖 1　節點動態調整緩衝區內的 description

有能力為較晚進入系統的新進節點找尋適合的父節點，符合條件的父節點須持有影片片頭，以支援新進節點錯失的影片片頭內容。

利用 MDC 的特性可使節點動態的調整緩衝區內所暫存的 description 數量，將 description 更有效率的運用，以提高節點利用率及伺服器負載能力。當緩衝區空間滿載，節點為避免影片片頭資訊的遺失及維持服務新進節點的能力，漸進式的丟棄緩衝區內的 description，利用執行 description 丟棄程序後所釋放出的緩衝區空間，儲存緩衝區內剩餘的 description 資源，增加緩衝區內暫存的影片片頭時間長度，延長節點服務新進節點的等待時間。

隨著動態調整緩衝區程序的執行，緩衝區中暫存的 description 數量遞減，使呈現不同的狀態，如圖 1 所示，假設 P_1 及 P_2 分別在第 1 分鐘及第 12 分鐘抵達系統，在圖 1(a)中，P_1 暫存 4 條 description 且仍然持有影片片頭的緩衝能力為 3 分鐘。因此在第 4 分鐘時，P_1 的緩衝區將會呈現滿載狀態，為保留影片片頭並等待新進節點的服務請求，節點可將目前緩衝區中的某一條 description 丟棄，圖 1(b)即為節點丟棄一條 description 後的狀態，由於釋放一條 description 的空間可用來儲存並延長緩衝區內剩餘的 3 條 description 暫存的時間長度，因此節點調整後的片頭長度增加為 4 分鐘。以此類推，如圖 1(d)所示，當緩衝區內只暫存一條 description 時，節點持有影片片頭的時間長度可延長至第 12 分鐘，當 P_2 在第 12 分鐘抵達時，由於 P_1 仍持有影片片頭，因此 P_2 可向 P_1 請求服務，剩下未滿足的觀看品質再由伺服器端進行服務。

在系統中會存在兩種類型的節點可執行動態調整的程序：

- **葉節點(leaf node)**。此類節點在系統中為最晚加入且尚未服務其他節點，緩衝區中仍有剩餘空間且持有影片片頭。
- **閒置節點(idle node)**。此類節點已服務其它節點，但其仍有剩餘的上傳頻寬。節點緩衝區中暫存的 description 狀態決定執行動態調整的能力，可區分為兩類：未上傳及已上傳。緩衝區中的未上傳 description 亦可定義為「閒置狀態的資源」。緩衝區未滿載且有 description 屬於閒置狀態是閒置節點的特性，因此，在執行動態調整時，須進一步考量緩衝區中的 description 是否為閒置狀態。由於非閒置 description 已上傳且服務其它節點，為避免影響子節點的 description 接收與影片觀看品質，故在執行 description 丟棄的程序時僅能針對閒置的 description 資源進行調整。

3.1　中央集權式管理

系統為有效的管理節點執行動態調整的程序，透過伺服器端實施中央控制及管理，降低新進節點向伺服器端請求的次數。節點執行動態調整緩衝區的程序時，必須考慮系統內 description 平衡分布的特性，以維持節點提供 *full-quality* 影片串流的能力。

3.1.1　基礎協定

在動態緩衝區的方法中，影片串流在傳送前須進行編碼，由伺服器端利用 MDC 編碼技術，將原始影片串流分割成為 m 條相同位元率(r_k)的 description。每條 description 都有固定編號 k, $1 \leqq k \leqq m$。節點(P_j)向系統請求服務之前，須提供必要的

相關資訊，再由伺服器端控制並觸發動態調整的相關作業程序，其區分如下：

- **Peer**。每個新進節點抵達系統時，必須主動提供相關的資訊，包括緩衝區大小(b_i)、所需的影片觀看品質(Q_i)及上傳頻寬(C_i)。
- **Server**。系統以伺服器端為中心，實行中央集權控管，建置 *Cache Information Table*(*CacheInfo Table*)以管理系統內執行動態調整的節點資訊，將各個節點的編號設為索引，記錄每個節點的狀態及相關資訊，其中包括節點的加入時間(t_i)， description 的持有狀況及數量(n_i)，並同時計算節點執行動態調整的時間(β_i)與剩餘頻寬。*CacheInfo Table* 內的資訊須持續追蹤及更新節點資訊，直到節點的緩衝區內不再保有影片片頭。

在伺服器端建置與維護 *CacheInfo Table* 的主要目的是為評估節點的服務能力及緩衝區滿載時間，在節點的緩衝區呈現滿載狀態之前，指定 description 的丟棄以平衡系統資源。伺服器端可利用 *CacheInfo Table* 內的資訊處理節點動態調整時的執行程序，其可區分為：

- **評估節點的服務能力**。一個具有服務能力的節點，亦可定義為執行動態調整的節點，其必須具備的條件為緩衝區中暫存大於一條的 description、持有影片片頭及至少足夠服務一條 description 的剩餘上傳頻寬，符合以上條件的節點將歸屬於可服務的節點集合(A)中。判斷節點服務能力的準則如公式(1)所示

$$A=\left\{P_i \middle| n_i>1 \text{ and } t_{now}-t_i<\frac{b_i}{n_i\times r_k} \text{ and } C_i\geq r_k\right\} \quad (1)$$

- **計算節點執行動態調整的時間**。伺服器端透過 *CacheInfo Table* 計算緩衝區暫存不同 description 數量的滿載時間長度，以 $\alpha_i(n_i)$表示，將節點的緩衝區滿載時間和加入時間點相加，即可得知節點執行動態調整的時間。如公式(2)所示，當節點的滿載時間與目前系統時間(t_{now})相符合時，節點會立即執行動態調整的程序，從緩衝區中選擇一條 description 進行丟棄。

```
/* Calculation of Available Set */
1    if n_i > 1 and t_now − t_i < b_i / (n_i × r_i) and C_i ≧ r_k then
2        A ← A ∪ P_i
3        else A ← A − P_i
4    end if
/* Calculation of a peer i's adjust time */
5    if  P_i ∈ A  then
6      For each peer P_i
7          for  n_i ← Q_i to 1
8            α_i(n_i) ← b_i / n_i×r_k
9            β_i ← t_i + α_i(n_i)
10             if  β_i = t_now  then
11                 n_i ← n_i - 1
                   Update β_i and n_i in CacheInfo table
12           else if n_i = 1 and β_i = t_now then
13               A ← A − P_i
14           end if
15   end if
```

圖 2 動態調整緩衝區演算法

$$\begin{cases}\alpha_i(n_i)=\dfrac{b_i}{n_i\times r_k}\\ \beta_i=t_i+\alpha_i(n_i)\end{cases} \quad (2)$$

圖 2 為伺服器端利用 *CacheInfo Table* 以評估節點是否具備執行動態調整的能力及計算動態調整時間的演算法。

3.2 資源平衡

此章節中將探討分群(Grouping)，其概念的提出是為了輔助緩衝區執行description丟棄的程序，以「集合」為單位使系統資源分配平衡化，並將系統內即將失去暫存影片片頭能力的節點優先分派

給新進節點作為父節點，善用系統所有節點的資源，且有利新進節點對於父節點的選擇，使其快速找到適合的父節點。

3.2.1 分群

若系統內的 description 資源被節點持有的數量不平衡，只要某一條 description 被執行動態調整之節點持有的數量為 0，則系統失去由節點提供 *full-quality* 的能力。此時，伺服器端為滿足新進節點對於觀看品質的需求，將繁複的處理無法由系統內節點服務的需求。若系統能有效規劃使節點持有的 description 資源維持平衡，則可降低透過系統內部節點服務新進節點的拒絕率，並滿足新進節點不同的觀看品質需求。

有鑒於上述之 description 資源平衡問題，在動態緩衝區方法中，節點執行 description 丟棄的程序時，將優先保留目前系統上最為欠缺的 description，換言之，系統目前被節點大量重覆持有的 description 是節點於丟棄程序時的優先考量。但由於節點緩衝區內所暫存的 description 數量及節點服務時間能力的不同，若忽略節點的特性，則將影響系統資源的平衡。假設目前系統內 *description 1* 被節點持有的數量最多，但 *description 1* 的資源集中於節點緩衝區只暫存一條 description 的情況，亦可表示為 $\alpha_i(1)$，則代表節點的服務能力即將結束，在下個時間點由節點提供 d_1 資源的服務能力將大為降低。因此，為避免不正確的系統資源估計，Grouping 將 n_i 作為節點的分群標準，平衡緩衝區內暫存相同 description 數量之節點持有的資源。

3.2.2 Description的丟棄

動態緩衝區方法中以緩衝區內暫存的description數量，即n_i，作為節點的分群準則，其所形成的集合以D_s表示，$1 \leqq s \leqq m$，目的為使持有description數量相同的節點資源均勻分布，平衡不同編號的description數量。因此，當節點執行動態調整時，應考慮調整後的集合中節點持有資源的數量以決定description的丟棄，優先選擇丟棄即將移至的集合中數量最多的description。

舉例來說，假設P_i緩衝區暫存4條description，即$n_i = 4$，則P_i目前是屬於「暫存4條description」集合中的節點，即D_4。當P_i觸發下次的動態調整時，其緩衝區內暫存的description數量將會減少一條，則P_i成為「暫存3條description」集合中的成員，即D_3。P_i於動態調整的description丟棄程序時，必須考慮「暫存n_i-1條description」的集合中，即D_3中節點的description資源持有狀況，再針對集合中資源數量為佔最多數量的description進行丟棄，保留較為欠缺的description，以平衡節點持有的description數量，使不同編號的description在集合中能呈現平衡狀態。

節點在進行調整及丟棄description之前，須透過*CacheInfo Table*得知節點服務的子節點數量(S_i)及己上傳的description集合(M_i)，以判斷節點是否為一個閒置節點。若一個節點緩衝區於滿載時間且在系統中屬於葉節點，表示為S_i=0或M_i=∅，則從目前持有的description集合(H_i)中選擇一條編號為k的description進行丟棄，其k是在D_s集合中被節點持有數量(R_k)最多的description編號，也就是說選擇D_s集合中數量最多的description丟棄。但若此節點已有服務其他節點，目前有緩衝區內中存在閒置資源並有剩餘上傳頻寬，表示為$S_i > 0$ 且$L_i \geqq 1$(H_i - $M_i \neq \varnothing$)且$C_i \geqq r_k$，則於系統中屬於閒置節點。其緩衝區會同時存在已上傳及未上傳的description，當節點緩衝區時間滿載時，閒置節點只會針對未上傳的description進行丟棄，圖3及圖4分別為葉節點與閒置節點執行description資源平衡的動態調整演算法。

3.2.3 資源利用最大化

延伸上述之 description 資源平衡的概念，動態緩衝區為提高節點利用率，更進一步考慮節點的服務時間。系統以緩衝區即將滿載的節點為服務新進節點的優先考量，目的為達到節點資源利用率最大

```
/* Leaf node drop description at adjust time */
For each Adjust Peer i in D_s
1   For k ← 1 to m
2     For each peer i in D_{s-1}
3     Find a description x subject to R_x ← MAX_{1≤i≤k} R_i
4       if β_i = t_now then
5         H_i ← H_i - x
6         n_i ← n_i -1
7         D_{s-1} ← D_{s-1} ∪ P_i
8       end if
```

圖 3　葉節點執行動態調整的資源平衡演算法

```
/* Idle node drop description at adjust time */
For each Adjust Peer i in D_s
1   if β_i = t_now then
2     if L_i = 1 then
3       Find a description x subject to x ∉ M_i
4         H_i ← H_i - x
5         n_i ← n_i -1
6         D_{s-1} ← D_{s-1} ∪ P_i
7     else if L_i > 1 then
8       For each Peer i in D_{s-1}
9         For each description
10          Find a description x subject to
              x∈(H_i - M_i) and R_x ← MAX_{1≤i≤k} R_i
11          H_i ← H_i – x
12          n_i ← n_i -1
13          D_{s-1} ← D_{s-1} ∪ P_i
14    end if
15  end if
```

圖 4　閒置節點執行動態調整的資源平衡演算法

化。如圖 5 所示，假設系統內存在 P_2 及 P_3 兩個節點，其緩衝區長度皆為 10 分鐘。P_2 的緩衝區暫存一條 description 的滿載時間為 3 分鐘後，P_3 的緩衝區暫存兩條 description 的滿載時間則為 2 分

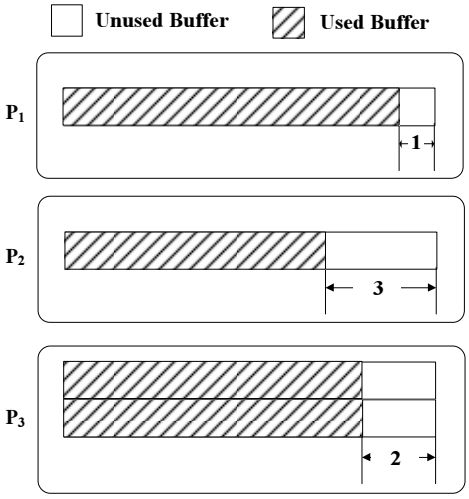

圖 5　P_1、P_2 及 P_3 是分屬於不同集合的節點

鐘後。若不考慮緩衝區內暫存 description 的數量，系統會誤判 P_3 為一個「緩衝區即將滿載的節點」，當有新進節點提出請求時，系統會指派 P_3 為父節點，P_2 的緩衝區空間長度剩餘 3 分鐘，即等待新進節點的最長時間為 3 分鐘。

然而，考慮節點緩衝區暫存的資源數量，由於 P_2 只暫存一條 description，即可視為無法執行動態調整的節點，其緩衝區滿載時間為 3 分鐘。 P_3 暫存兩條 description 仍可進行一次動態調整，其實際的緩衝區滿載時間為 12 分鐘，換句話說，系統等待新進節點的時間可延長至 12 分鐘。因此，當有新進節點抵達時，系統將指派 P_2 為父節點，以增加系統等待新進節點加入的等待時間。

Grouping 中以節點緩衝區暫存的 description 數量為分群準則，並以緩衝區滿載時間決定節點服務順序，強調節點之間的相異性。同時考慮節點緩衝區中暫存的 description 數量及節點的動態調整時間，在分群的概念中，以緩衝區內暫存的 description 數量作為分群的準則。在此，為更進一步的達到節點資源利用最大化的目標，伺服器端透過 *CacheInfo Table* 可將各個集合中節點的動態調整時間進行升冪排序，其主要目的為將節點動態調整時間越接近系統目前時間且緩衝區中暫存 description 數量少的節點優先配置給新進節點作為父節點。如圖 5 所示， P_1、P_2 及 P_3 分別為 *CacheInfo Table* 中所記錄的節點，依 Grouping 的概念，緩衝

/* A new peer arrived selecting a parent*/

For a newly arrived peer i

1 Repeat

2 if $A \neq \varnothing$ and $P_i \in A$

3 For each peer x in D_s subject to $\beta_x \leftarrow \underset{1 \le j \le i}{MIN}\ \beta_j$

4 if $n_x = 1$ and $H_x - H_i \neq \varnothing$ then

5 $H_i \leftarrow H_i \cup H_x$

6 else if $n_x > 1$ and $H_x - H_i \neq \varnothing$ then

7 Find a description c subject to $R_c \leftarrow \underset{1 \le k \le m}{MIN}\ R_k$

8 $H_i \leftarrow H_i \cup c$

9 $Q_{i+1} \leftarrow Q_i + 1$

10 $n_i \leftarrow n_i + 1$

11 end if

12 end if

13 Until $Q_i = 0$ or $A = \varnothing$

圖 6 新進節點挑選適合的父節點

區內暫存的 description 數量決定節點所屬的集合，即 P_1 及 P_2 同屬於 D_1 集合中的節點，P_3 則屬於 D_2。當一個新進節點抵達系統時，伺服器端會優先考慮 D_1 集合中的節點，並根據 *CacheInfo Table* 中的節點緩衝區滿載時間，選擇即將失去服務能力的節點作為新進節點的父節點，即 P_1。

伺服器端根據節點的條件不同進行分群，並優先考慮緩衝區快滿載且即將失去持有影片片頭能力的節點，結合 Grouping 及資源利用最大化的概念，可避免系統誤判可服務節點之滿載時間，並使節點服務利用率最大化。圖 6 即為新進節點挑選父節點的演算法。

四、模擬實驗

4.1 實驗環境設定

在此章節中將針對動態緩衝區的模擬實驗結果進行分析及評估。本模擬實驗中，假設伺服器端

表 1 模擬實驗的初步參數設定

參數	數值
伺服器頻寬(description)	5000
影片長度(分鐘)	120
Description 的數量	16
到達率(節點數量/分鐘)	0-100
緩衝區大小(分鐘)	32

的頻寬為 5000 條 description，影片長度為 120 分鐘，並使用 MDC 將原始影片串流進行編碼。由於動態緩衝區是建置於同儕式網路上的隨選視訊服務，其透過節點將影片片頭暫存，以服務新進節點，故節點的緩衝區空間的大小對隨選視訊服務會產生直接的影響。

本實驗中將節點的緩衝區空間是以亂數產生服從均勻分配，其範圍為[2-32]，其初始設定為 32 分鐘，並依節點請求的觀看品質而有所變動。若節點對影片觀看品質的需求為 *full-quality*，即為 16 條 description，則緩衝區暫存 16 條 description 的緩衝長度為 2 分鐘，若請求的影片觀看品質為 8 條 description，則緩衝區暫存 8 條 description 的長度為 4 分鐘，以此類推。節點到達率為 poisson 分配，λ 為每分鐘 0-100 個節點，且節點對於影片的觀看品質則為常態分配，N(9,3)。此外，節點的上傳頻寬及下載頻寬相等，也就是說節點所接收的 description 數量會與貢獻的 description 數量相等。表 1 為本模擬實驗的相關參數設定。

在本實驗中，提出不同方法針對其伺服器端的頻寬消耗進行探討，分別為 CoopNet、動態緩衝區、動態緩衝區－Sequence 以及動態緩衝區－Random。

- **CoopNet**。利用 MDC 技術的特性，改善節點到達率過於密集而造成伺服器超載的情況，使系統整體有較佳的擴充性，但 CoopNet 的方法中不會對節點的緩衝區執行動態調整。
- **動態緩衝區**。節點執行動態調整時須考慮系統中 description 的資源平衡。

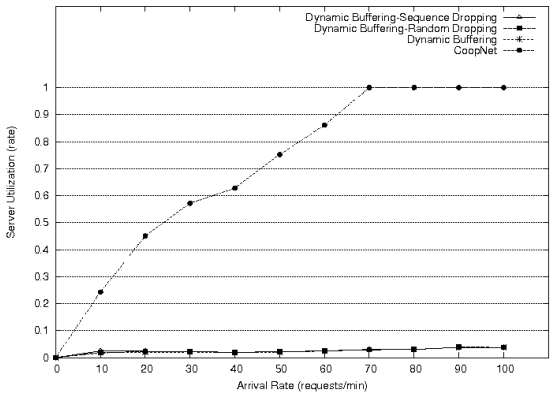

圖 8 不同節點到達率對伺服器頻寬的耗用 (Buffer Size=32 min)

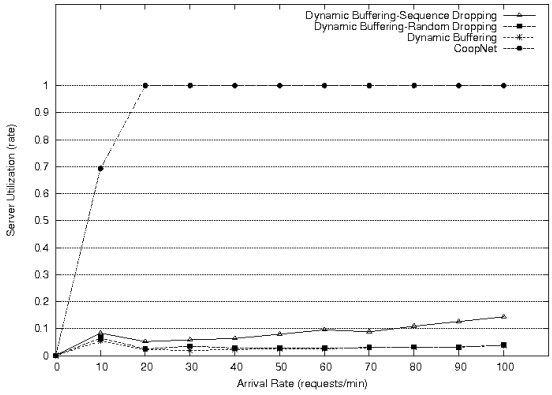

圖 9 不同節點到達率對伺服器頻寬的耗用 (Buffer Size=10 min)

- **Sequence**。此方法為動態緩衝區的的延伸但不考慮 description 資源平衡的概念，節點執行動態調整時，選擇欲丟棄的 description 是依照編號的順序。
- **Random**。此方法為動態緩衝區的延伸，不考慮 description 資源平衡的概念，節點執行動態調整時，選擇欲丟棄的 description 是以亂數隨機產生欲丟棄的 description 編號。

4.2 效能評估

在效能評估與分析中，藉由節點的緩衝區大小的改變，觀察並探討不同方法對於伺服器端所造成的負載情形。

圖 8 中設定不同的節點到達率，分別對四種方法進行模擬。當節點的到達率越高，則伺服器的負載亦隨之增加，從圖 8 中明顯可看出 CoopNet 中，提高節點到達率，伺服器端的頻寬使用率呈現上升狀態，當節點到達率為 70 時，伺服器已無多餘頻寬可服務新進節點的需求，因為每分鐘到達的節點數量太多，且系統中的節點服務能力受限於緩衝區空間的大小，使得系統內的節點無法服務新進節點，故當節點到達率位於 70~100 的區間內伺服器端頻寬的耗用皆是持平於最高點。反之，在動態緩衝區中，節點到達率的增加，只造成伺服器端頻寬耗用上些微的變化，且增加的幅度並不會造成伺服器端發生負載過重的問題。

當節點到達率為 10 時，CoopNet 的伺服器頻寬耗用為 24.28%，動態緩衝區則為 1.72%，當節點到達率為 100 時，CoopNet 的伺服器頻寬耗用為 100%，動態緩衝區則為 3.76%，相較於 CoopNet，動態緩衝區具有良好的系統擴充性。換言之，即動態緩衝區，透過節點動態調整緩衝區的 description 資源，可降低隨選視訊服務提供冷門及熱門影片所造成伺服器負載過重的問題，提高服務新進節點的能力。

圖 8 中同時可觀察到 Sequence 及 Random 的方法相當接近動態緩衝區，其原因是在動態緩衝區中資源平衡主要是支援節點執行動態調整時丟棄 description 的程序，目的改善被節點持有 description 的數目使呈現平衡。然而，在此初步設定節點的緩衝區的最大長度為 32 分鐘，節點有充裕的時間等待新進節點的加入，節點執行動態調整的時間拉長，資源平衡在此情況中所發揮的效用並不顯著。

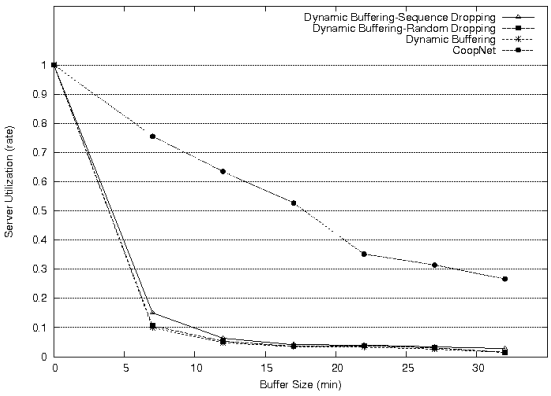

圖 10 不同節點緩衝區大小對於伺服器頻寬的耗用(Arrival Rate=10)

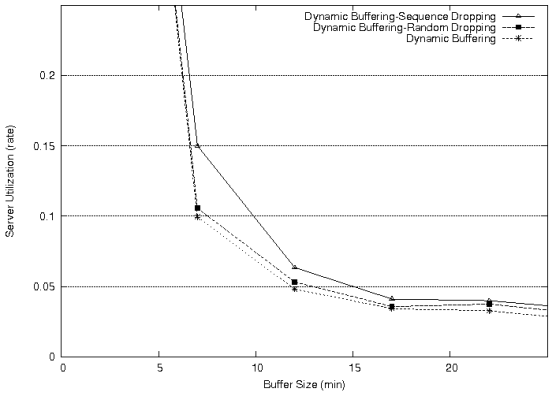

圖11 不同description丟棄方法對於伺服器頻寬的耗用(Arrival Rate=10)

為了區分 Sequence 及 Random 不同於資源平衡的概念，圖 9 中設定不同的節點到達率，觀察對伺服器的頻寬耗用情形，在此將節點的緩衝區的最大長度設為 10 分鐘。在 CoopNet 中，由於節點緩衝區暫存影片片頭的長度不足等待新進節點的加入，新進節點轉而向伺服器請求，因此，當節點到達率為 20 時，系統即無法服務任何新進節點的請求。當節點的緩衝時間較短時，為了保留影片片頭資訊，節點進行動態調整，由於緩衝區空間較小，故緩衝區的調整時間相當密集。在 Sequence 中由於未考慮資源平衡，系統內的被節點持有的資源產生不平衡的現象，當新進節點發出服務請求時，系統內節點無法提供多樣化的資源。在 Random 中，採用亂數隨機的方式決定編號，伺服器頻寬的耗用介於動態緩衝區及 Sequence 之間。

當節點到達率為 10 時，CoopNet 的伺服器頻寬耗用為 69.26%，Sequence 為 8.32%，Random 為 6.48%，動態緩衝區則為 5.38%。當節點到達率為 100 時，CoopNet 的伺服器頻寬耗用為 100%，Sequence 為 14.38%，Random 為 3.84%，動態緩衝區則為 3.8%。相較於 CoopNet，動態緩衝區能有效改善節點資源的配置，以維持資源平衡，在降低節點的緩衝能力且到達率較低的情況下，仍可盡力滿足新進節點對於影片觀看品質的不同需求。

這是由於 MDC 和動態緩衝區互相搭配的綜效，舉例來說，當某一個節點握有 12 條 description，而它的緩衝區有暫存 32 分鐘影片片頭的能力，在 CoopNet 中此節點只能保有影片片頭 2 分鐘，在第 3 分鐘時，節點已沒有持有影片片頭的能力(緩衝區空間須 36 分鐘)，如此一來系統並無法充分利用節點的緩衝能力，然而在動態緩衝區中，則可透過丟棄 description，減少緩衝區中暫存的 description，釋放出多餘的儲存空間，延伸緩衝區內可以保有影片片頭的時間。動態緩衝區用漸進式的調整緩衝區內的 description 數量，以延伸影片片頭長度，避免節點的緩衝區空間內一次全然沒有保留任何一條 description 片頭。即使當節點到達率增加時，亦可透過節點動態調整及資源平衡，對於伺服器端的負載有明顯的改善。

圖 10 設定節點到達率為 10，以觀察在不同的緩衝區長度下，節點的請求對伺服器的頻寬耗用情形，在此將節點緩衝區的最大長度分別設定為 7、12、17、22、27 及 32 分鐘。

如圖 10 所示，當節點緩衝區長度越長，增加等待新進節點的時間，影片片頭可調整的長度亦越長，相對提高由系統內節點服務新進節點的機率，進而減少新進節點向伺服器端請求服務的次數。圖 11 為圖 10 中動態緩衝區之不同丟棄 description 方式的詳圖，其顯示 Sequence、Random 及動態緩衝

區三種丟棄方法的差異，當節點緩衝區較小時，動態緩衝區中提出的資源平衡有較低的伺服器端頻寬耗用。

當節點緩衝區的最大長度為 7 分鐘時，CoopNet 的伺服器頻寬耗用為 75.52%，Sequence 為 15.02%，Random 為 10.56%，動態緩衝區則為 9.94%。當節點緩衝區的最大長度為 32 分鐘時，CoopNet 的伺服器頻寬耗用為 26.64%，Sequence 為 2.8%，Random 為 1.52%，動態緩衝區則為 1.46%。當節點緩衝區空間愈大時，動態緩衝區會降低對伺服器頻寬的消耗，這是由於節點的緩衝區有足夠的空間拉長影片片頭的時間，但節點緩衝區空間的大小是系統難以預測的問題。

相較於 CoopNet，動態緩衝區更能適應各種不同的節點條件，在節點緩衝區儲存空間不足的情況下，仍有效的降低伺服器負載，動態緩衝區具有較高的系統擴展性，且有彈性的適應不同的網路環境及突破節點能力限制對伺服器的可能帶來的影響。

五、結論

本篇論文中提出建置於同儕式網路上的隨選視訊服務之動態緩衝區的技術。系統內持有影片片頭的節點，即具有服務影片片頭能力的節點，動態調整緩衝區內暫存的影片片頭長度，其調整的程序主要是為了提升節點服務能力，且改善系統內不同的節點到達率對於伺服器端所造成的頻寬負載量。動態緩衝區可適應隨選視訊服務系統中的使用者對不同熱門程度的影片需求，當使用者對影片內容的需求較極端時，即熱門與冷門電影，節點透過動態的調整緩衝區內所暫存的 description 內容，延長節點等待服務新進節點的可容忍時間，改善系統的整體效能，且亦增加系統擴充性。

本篇論文的研究限制為適用於建置在同儕式網路上之隨選視訊服務，其透過節點相互合作以取得影片內容，因此當系統發生節點離開的事件時，將造成其他節點的觀看品質受到影響，系統必須有效的處理因節點離開而導致其他節點觀看中斷或無法觀看的情況，因此錯誤回覆將是本篇論文未來研究將探討的議題。

參考文獻

[1] A. Dan, D. Sitaram , and P. Shahabuddin, “Batching ： Scheduling policies for an on-demand video server with batching, “ Proc. of ACM Multimedia, pp.15-23, 1994.

[2] A. Rowstron and P. Druschel, “ Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer system, ” Proc. of ACM Distributed Systems Platforms(Middleware), pp. 329-350, 2001.

[3] E. Kusmierek, Y. Dong, and D.H.C. Du, ” Loopback : Exploiting Collaborative Caches for Large-Scale Streaming, ” IEEE Transactions on Multimedia, vol. 8, no. 2, pp. 233 - 242, 2006.

[4] I. Lee, Y. He, and L. Guan., “ Centralized P2P Streaming with MDC, ” Proc. of IEEE Multimedia Signal, pp.1-4, 2005.

[5] K. A. Hua, Y. Cai, and S. Sheu, “Patching: a multicast technique for true video-on-demand services , “ Proc. of ACM Multimedia, pp.191-200, 1998.

[6] L. S. Juhn and L. M. Tseng, “Harmonic broadcasting for video-on-demand service, ” IEEE Transactions on vol. 43, no. 3, pp.268 - 271, 1997.

[7] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron and A. Singh, “ SplitStream: high-bandwidth multicast in cooperative environments, ” Proc. of ACM Symposium on Operating System Principles, pp.298-313, 2003.

[8] M. Castro, P. Druschel, A. M. Kermarrec, and A. Rowstron, “ SCRIBE : A large-scale and decentralized application-level multicast infrastructure, “ IEEE Journal on Selected Areas

in Communication, vol. 20, no. 8, pp.1489-1499, 2002.

[9] S. Kulkarni and J. Markham, " Split and merge multicast : live media streaming with application level multicast, " Proc. of IEEE Communications, pp.1292-1298, 2005.

[10] S. Viswanathan and T.Imielinski, " Metropolitan area video-on-demand service using pyramid broadcasting, " ACM Multimedia Systems, vol. 4, no. 4, pp.197-208, 1996.

[11] T. T. Do, K. A. Hua, and M. A. Tantaoui, " P2VoD : providing fault tolerant video-on-demand streaming in peer-to-peer environment, " Proc. of IEEE Communications, pp.1467-1472, 2004.

[12] V. N. Padmanabhan, H. J. Wang, and P. A. Chou , " Resilient peer-to-peer streaming, " Proc. of IEEE Network Protocols, pp.16-27, 2003.

[13] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking, " Proc. of ACM NOSSDAV, pp.177-186, 2002.

[14] W. H. Ma and D. H. C. Du, " Design a progressive video caching policy for video proxy servers, " IEEE Multimedia, vol. 6, no. 4, pp.599-610, 2004.

[15] Y. Cui and K. Nahrstedt, " Layered peer-to-peer streaming, " Proc. of ACM NOSSDAV, pp.162-171, 2003.

[16] Y. Guo, K. Suh, J. Kurose, and D. Towsley," P2Cast: peer-to-peer patching scheme for VoD service, " Proc. of ACM WWW, pp.301-309, 2003.